Capstone Design Final Report

AL AKHAWAYN UNIVERSITY
SCHOOL OF SCIENCE AND ENGINEERING

AUI INTERNAL SERVICES

Salma El Fatihi

Supervised by: Dr Bouchaib Falah
AL AKHAWAYN UNIVERSITY

AL AKHAWAYN UNIVERSITY IN IFRANE

SCHOOL OF SCIENCE & ENGINEERING

AUI Internal Services Application

AUI INTERNAL SERVICES

EGR 4402: Capstone Design

20th April 2015

Salma El Fatihi

Supervised by Dr Bouchaib Falah
ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Bouchaib Falah for all the help and feedback he gave me within the progress of my diaries, my reports and my application, as well as Al Akhawayn University for the skills, knowledge and both the theoretical and practical trainings it provided me all along these four years. They were of great help in terms of preparing me to conduct and submit a final project individually. Finally, I would also like to thank Mrs. Mounia Aaquil who works at the Business Office and who was my client. Indeed, she was always helpful and ready to answer my questions and concerns regarding the application.
# CONTENTS

I Introduction .................................................................................................................. 1  
1.1 Scope of Project............................................................................................ 1  
1.2 Definitions, Acronyms and Abbreviations....................................................... 2  
1.3 Overview of Document.................................................................................... 2  

2 Requirements Specification...................................................................................... 3  
2.1. Functional Requirements ............................................................................. 3  
2.2. Non Functional Requirements........................................................................ 10  
2.2.1. Product Requirements.............................................................................. 10  
2.2.1.1. Usability Requirements................................................................. 10  
2.2.1.2. Reliability Requirements.............................................................. 10  
2.2.1.3. Efficiency Requirements................................................................ 10  
2.2.1.3.1. Performance........................................................................... 10  
2.2.1.3.2. Space.................................................................................. 10  
2.2.1.4. Portability Requirements................................................................ 10  
2.2.2. Organizational Requirements................................................................. 10  
2.2.2.1. Delivery Requirements.................................................................... 10  
2.2.2.2. Implementation Requirements...................................................... 11  
2.2.2.3. Standards Requirements................................................................ 11  
2.2.3. External Requirements............................................................................ 11  
2.2.3.1. Interoperability Requirements....................................................... 11  
2.2.3.2. Ethical Requirements – STEEPLE ANALYSIS............................ 11  
2.2.3.3. Legislative Requirements.............................................................. 11  
2.2.3.3.1. Security............................................................................... 11  
2.2.3.3.2. Safety............................................................................... 11  
2.2.3.3.3. Privacy............................................................................. 11  

3 Feasibility Study.................................................................................................... 12  
3.1. Technology Enablers..................................................................................... 12  
3.1.1. Platform......................................................................................... 12  
3.1.2. Database Server............................................................................. 12  
3.1.3. Programming Languages..................................................................... 12  
3.1.4. Text Editor................................................................................... 13  
3.2. Schedule..................................................................................................... 14  

4 Design and Analysis............................................................................................ 15  
4.1. Entity Relationship Diagram....................................................................... 15  
4.2. Use Case Diagram...................................................................................... 16  
4.3. Examples of Sequence Diagrams................................................................. 17  
4.3.1. Authentication Sequence Diagram.................................................... 17  
4.3.2. Add Apartment Sequence Diagram.................................................. 17  
4.3.3. Update Apartment Sequence Diagram............................................. 18  
4.3.4. Delete Apartment Sequence Diagram............................................. 18  

5 Development and Implementation..................................................................... 19  
5.1. Data Model Tier....................................................................................... 19  
5.2. Business Tier......................................................................................... 19  
5.3. Web Tier............................................................................................... 23  
5.4. Screenshots of the Application.................................................................. 24  

6 Conclusion......................................................................................................... 41  
References........................................................................................................... 42  
Appendix A........................................................................................................... 43
ABSTRACT

This capstone project is about developing a web application for the Business Office of Al Akhawayn University. This application will handle mainly the internal services of the university, especially the interactions between the Business Office, the Housing Department and the Faculty members.

The first phase of this project consists of the requirements specification which was achieved through meetings with the Business Office in order to clarify the meaning of “Internal Services”. Indeed, the client stated that these two words include the assignment of different apartments by the Housing Department to Faculty members, and the charges of electricity and water bills by the Business Office to those Faculty members. Therefore, the purpose of this project is to combine all these services, which are basically achieved through emails between the three parts, into one formal web application in order for everyone to be involved.

Hence, the requirements specifications of this report will include mainly the functional and non-functional requirements. Then comes the second part of the project which will consist of the feasibility study. In this phase, a detailed report of which IDE, programming languages and servers were used will be given in order to justify how the requirements were correctly met.

Before starting the implementation phase, it is mandatory to go through the design and analysis part, in which an entity relationship diagram, a use case diagram and examples of sequence diagrams will be made. At last will come the implementation part which will be justified through screenshots of the main functionalities of the web application in addition to some important parts of the code.

Last but not least, this application will contribute to helping the Business Office, the Housing Department and the Faculty members interact in a clear and concise environment, in which data is safe and preserved from any loss, and information is transparent and available to any part at any time. Finally, the application will contain a list of references to all the sources and documentation used. In addition, it will also follow the common standards and regulations of the capstone project and the IEEE SRS templates.
1 Introduction

As a student and as part of the requirements of the engineering bachelor’s degree, I was prompted to take a Capstone Project in order to make into practice the skills and the theoretical knowledge I have been accumulating during the last four academic years.

My project is called “AUI Internal Services Application” and it is a web application that combines the work of the Housing Department, the Business Office and the Faculty members. When Faculty members join the AUI community, they are assigned apartments by the Housing Department, which communicates the assignment information to the Business Office that charges the Faculty members with water and electricity bills. This procedure is fully done through emails and papers, therefore, this application has the purpose of automatizing all the work and making it accessible to the three parts through a single interface.

This web application will enable both the Housing Department and the Business Office to enter, modify, search and delete their data and information in a complementary manner, which will be available for the Faculty members to check and print if necessary. This document provides a detailed reporting of the different steps through which the progress of the implementation of my web application was done.

1.1 Scope of Project

As mentioned before, the web application to be produced is named “AUI Internal Services Application”. It will combine the work of the Housing Department, the Business Office and the Faculty members in a complementary manner. It will enable the first two departments to enter, modify, search and delete their data and information related to the apartments’ assignments and the water and electricity bills of each Faculty member, who will also be able to check this information and print it if necessary. It is beneficial because it will preserve from the use of papers and then reduce the costs, in addition to the fact that it will eliminate the classic communication through emails, which is not really efficient nor organized, in addition to the fact that it does not guaranty for the data not to be lost, either by being deleted or classified as junk emails by mistake, or by hiding within the massive amount of different emails from the same account.
1.2 Definitions, Acronyms, and Abbreviations

SRS: Software Requirement Specification
IEEE: Institute of Electrical and Electronic Engineers
SSD: Software Specification Document
AUI: Al Akhawayn University in Ifrane
ERD: Entity Relationship Diagram
RDBMS: Relational Database Management System

1.3 Overview of Document

As specified by my supervisor, this software specification document, SSD, will follow the IEEE SRS and will include the following sections in their respective order: a detailed introduction containing the scope of the project, the glossary and the overview of the document; a requirements specification section containing the functional requirements, the non-functional requirements, the product requirements (usability – reliability – efficiency [performance and speed] - portability), the organizational requirements (delivery – implementation – standards) and the external requirements (interoperability – ethical or STEEPLE analysis – legislative [security – safety – privacy]); a design and analysis part containing the entity relationship diagram, ERD, the use case diagram and examples of some sequence diagrams; an implementation part containing screenshots of important parts of the code and screenshots of the main functionalities of the web application; a conclusion section and finally some references.
2 Requirements Specification

During the first meeting with the client, I asked questions for more understanding and clarification, and I also took notes and gathered all the data and documents I judged to be necessary and helpful for producing this application. Some of these documents were real water and electricity bills and copies of emails that were sent to faculty members containing their charges (see Appendix A). Consequently, most of the time, the requirements were getting clearer throughout the progress of the work, and in case they did not, I decided to reschedule another meeting with the client. The following list shows the requirements separated into modules. Regarding the Account Management module, my client specifically required that the functionalities of this module should not be included in the application, but they rather be used by the database administrator for security reasons. This administrator will be chosen among the Housing Department or the Business Office in order to avoid any external user non-belonging to the three departments of the application to access it.

2.1 Functional Requirements

2.1.1 Account: Create new account

2.1.1.1 Purpose

This is not a function implemented in the database because the client wanted to choose a database administrator to be responsible for creating accounts, directly from the database, so that no external user non-belonging to the three departments of the application could access it.

2.1.1.2 Input

The database administrator shall input all the account information (username and password).

2.1.1.3 Output

The information added shall be displayed in the database.

2.1.2 Account: Update account

2.1.2.1 Purpose

This is not a function implemented in the database because the client wanted to choose a database administrator to be responsible for updating accounts, directly from the database, so that no external user non-belonging to the three departments of the application could access it.
2.1.2.2 Input
The database administrator shall update all information desired to be changed (username and password).

2.1.2.3 Output
The updated information shall be displayed in the database.

2.1.3 Account: Delete account
2.1.3.1 Purpose
This is not a function implemented in the database because the client wanted to choose a database administrator to be responsible for deleting accounts, directly from the database, so that no external user non-belonging to the three departments of the application could access it.

2.1.3.2 Input
The database administrator shall delete all information desired to be suppressed (username and password).

2.1.3.3 Output
The deleted information shall disappear from the database.

2.1.4 Apartment: Add apartment
2.1.4.1 Purpose
The housing officer shall be able to add a new apartment.

2.1.4.2 Input
The housing officer shall input all the information related to the apartment: building, number, ONE contract etc.

2.1.4.3 Output
The added apartment shall be displayed in the list of apartments.

2.1.4.4 Exception
An exception shall be displayed if the housing officer presses submit with empty fields.

2.1.5 Apartment: Update apartment
2.1.5.1 Purpose
The housing officer shall be able to update an existing apartment.

2.1.5.2 Input
The housing officer shall input all the new information related to the apartment desired to be changed.
2.1.5.3 **Output**
The updated apartment shall be displayed in the list of apartments.

2.1.5.4 **Exception**
An exception shall be displayed if the housing officer presses submit with empty fields.

2.1.6 **Apartment: Delete apartment**

2.1.6.1 **Purpose**
The housing officer shall be able to delete an existing apartment.

2.1.6.2 **Input**
The housing officer shall click on the apartment desired to be removed.

2.1.6.3 **Output**
The deleted apartment shall disappear from the list of apartments.

2.1.7 **Apartment: Search apartment**

2.1.7.1 **Purpose**
The housing officer shall be able to search an existing apartment.

2.1.7.2 **Input**
The housing officer shall enter any information related to the apartment desired to be searched.

2.1.7.3 **Output**
The searched apartment shall appear from the list of apartments.

2.1.7.4 **Exception**
an exception shall be displayed if the apartment does not exist.

2.1.8 **Apartment: Free apartment**

2.1.8.1 **Purpose**
The housing officer shall be able to free an apartment from his/her occupying faculty member.

2.1.8.2 **Input**
The housing officer shall click on the apartment desired to be freed.

2.1.8.3 **Output**
The freed apartment shall appear in the apartment list with value “0”.

2.1.8.4 **Exception**
An exception shall be displayed if the housing officer tries to free an apartment which is already free.
2.1.9 Faculty: Add faculty

2.1.9.1 Purpose
The housing officer shall be able to add a new faculty member.

2.1.9.2 Input
The housing officer shall enter the information of the faculty member, such as first name, last name, etc.

2.1.9.3 Output
The added faculty member should appear in the faculty list.

2.1.9.4 Exception
An exception shall be displayed if the housing officer presses submit with empty fields.

2.1.10 Faculty: Update faculty

2.1.10.1 Purpose
The housing officer shall be able to update an existing faculty member.

2.1.10.2 Input
The housing officer shall enter the new information of the faculty member desired to be changed.

2.1.10.3 Output
The updated faculty member should appear in the faculty list.

2.1.10.4 Exception
An exception shall be displayed if the housing officer presses submit with empty fields.

2.1.11 Faculty: Delete faculty

2.1.11.1 Purpose
The housing officer shall be able to delete an existing faculty member.

2.1.11.2 Input
The housing officer shall click on the faculty member to be deleted.

2.1.11.3 Output
The deleted faculty member should disappear from the faculty list.

2.1.12 Faculty: Search faculty

2.1.12.1 Purpose
The housing officer shall be able to search an existing faculty member.
2.1.12.2 Input
The housing officer shall enter any information related to the faculty member to be searched.

2.1.12.3 Output
The searched faculty member should appear from the faculty list.

2.1.12.4 Exception
An exception shall be displayed if the housing officer searches for a faculty member that does not exist.

2.1.13 Charges: Add charge
2.1.13.1 Purpose
The business officer shall be able to add a new charge.

2.1.13.2 Input
The business officer shall enter the information related to the charge, such as the amount due, the amount paid, etc.

2.1.13.3 Output
The added charge should appear in the charges list.

2.1.13.4 Exception
An exception shall be displayed if the business officer presses submit with empty fields.

2.1.14 Charges: Update charge
2.1.14.1 Purpose
The business officer shall be able to update an existing charge.

2.1.14.2 Input
The business officer shall enter the new information related to the charge desired to be changed.

2.1.14.3 Output
The updated charge should appear in the charges list.

2.1.14.4 Exception
An exception shall be displayed if the business officer presses submit with empty fields.

2.1.15 Charges: Delete charge
2.1.15.1 Purpose
The business officer shall be able to delete an existing charge.
2.1.15.2 Input
The business officer shall click on the charge to be deleted.

2.1.15.3 Output
The deleted charge should disappear from the charges list.

2.1.16 Charges: Search charge
2.1.16.1 Purpose
The business officer shall be able to search an existing charge.

2.1.16.2 Input
The business officer shall enter any information related to the charge to be searched.

2.1.16.3 Output
The searched charge should appear from the charges list.

2.1.16.4 Exception
An exception shall be displayed if the business officer tries to search for a charge that does not exist.

2.1.17 Charges: View apartment
2.1.17.1 Purpose
The business officer shall be able to view apartments added by housing officers.

2.1.17.2 Input
The business officer shall click on the apartments list.

2.1.17.3 Output
The existing apartments shall be displayed in the apartments list.

2.1.17.4 Exception
An exception shall be displayed if no apartment already exists.

2.1.18 Charges: View faculty members
2.1.18.1 Purpose
The business officer shall be able to view faculty members added by housing officers.

2.1.18.2 Input
The business officer shall click on the faculty list.

2.1.18.3 Output
The existing faculty members shall be displayed in the faculty list.
2.1.18.4 Exception
An exception shall be displayed if no faculty member already exists.

2.1.19 Bill: View bill
2.1.19.1 Purpose
The faculty member shall be able to view generated water/electricity bills.

2.1.19.2 Input
The faculty member shall click on the water/electricity bill.

2.1.19.3 Output
The water/electricity bill shall be displayed in the bill list.

2.1.19.4 Exception
An exception shall be displayed if no water/electricity bill is already generated.

2.1.20 Bill: Print bill
2.1.20.1 Purpose
The faculty member shall be able to print generated water/electricity bills.

2.1.20.2 Input
The faculty member shall click on the print button of the desired bill to be printed.

2.1.20.3 Output
The printing settings page should be displayed.

2.1.21 User: Update profile
2.1.21.1 Purpose
The user shall be able to update his/her profile.

2.1.21.2 Input
The user shall enter new information such as first name, email, etc.

2.1.21.3 Output
The updated information shall be displayed in the user profile.

2.1.21.4 Exception
An exception shall be displayed if the user presses submit with empty fields.

2.1.22 User: Change pass
2.1.22.1 Purpose
The user shall be able to change the password of his/her account.

2.1.22.2 Input
The user shall enter the old and new passwords and confirm the new one.
2.1.22.3 Output
The user shall be able to login with the new password.

2.1.22.4 Exception
An exception shall be displayed if the user presses submit with empty fields.

2.2. Non Functional Requirements

2.2.1 Product Requirements

2.2.1.1 Usability Requirements
The system shall be easy to use with a user-friendly interface.

2.2.1.2 Reliability Requirements
The system shall be reliable when it comes to performing the required functions.

2.2.1.3 Efficiency Requirements

2.2.1.3.1 Performance
The system shall be able to perform all its functionality under specific conditions. Therefore, it must be robust with minimum breakdowns. The time of execution is less than 10 seconds.

2.2.1.3.2 Space
The database used should be able to contain all the system data and information required. The space the application shall not exceed 4340 Kb, with the database file included.

2.2.1.4 Portability Requirements
The system shall be compatible with as many platforms as possible.

2.2.2. Organizational Requirements

2.2.2.1 Delivery Requirements
The system was delivered by April 27th, 2015.
2.2.2.2. Implementation Requirements

The system shall be implemented using PHP, CSS, HTML and JavaScript languages. As a text editor, Sublime Text 2 will be used and as a RDBMS, MySQL will be used on the WampServer platform.

2.2.2.3. Standards Requirements

The application shall respect the high professional IEEE standards for software requirement specification, by following the template provided by my supervisor.

2.2.3. External requirements

2.2.3.1. Interoperability Requirements

The application should be able to interact and communicate with other specified systems like the MySQL Database.

2.2.3.2. Ethical Requirements – STEEPLE ANALYSIS

The application shall maintain integrity and independence along with the support of the human values.

2.2.3.3. Legislative Requirements

2.2.3.3.1. Security

The application shall be protected from any external attack.

2.2.3.3.2. Safety

The application shall not damage the hardware, or the user or the environment.

2.2.3.3.3. Privacy

The information of the users shall respect the confidentiality of its clients.
3 Feasibility Study

After having set the requirements of the system, the next step was to decide about the technology enablers. The system implemented is to be used first by the Housing Department in order to make apartment assignments for Faculty members, then the Business Office will check the updates on the application and charge the same Faculty members with their related water and electricity bills. Finally, the Faculty members will be able to view their bills online and print them if necessary. As I am mostly familiar with the WampServer platform, I decided to use it for the implementation of this web application. Concerning the other technology enablers, this feasibility study has been conducted in order to justify which ones are going to be used.

3.1 Technology Enablers

3.1.1 Platform

WampServer is a platform for Web development of type WAMP, enabling to run PHP scripts locally. WampServer is not in itself a program, but an environment with two servers (Apache and MySQL), a shell script (PHP) and phpMyAdmin for web administration of MySQL databases [1].

3.1.2 Database Server

MySQL will be used as a database server for its numerous advantages. MySQL is very popular with web applications. It provides an open source server with a workbench GUI that facilitates its use. The ease of integration with different languages is another advantage of MySQL among many others [2].

3.1.3 Programming Languages

**PHP**

PHP is one the most famous general-purpose scripting languages used for web development. Some of its advantages are flexibility, speed and pragmatism. It can be used to develop everything ranging from blogs to very popular websites such as Facebook [3].
**HTML 5**

The Hypertext Markup Language, commonly abbreviated as HTML, is the data format that was designed to display web pages. It is a markup language that is used for writing hypertext, which explains the origin of its name. HTML also allows semantically structuring and reshaping the content of the pages, in addition to including multimedia resources such as images, entry forms and computer programs [4].

**CSS 3**

CSS (Cascading Style Sheets) is a computer language used to describe the presentation of HTML and XML documents. Standards defining CSS are published by the World Wide Web Consortium (W3C). Introduced in the mid-1990s, CSS is commonly used in web design and well supported by web browsers in the 2000s [5].

**JavaScript**

JavaScript is an open scripting language developed by Netscape for the purpose of developing interactive web applications. As its name suggests, it shares a lot of structures of features of the Java language, and it can also interact with HTML source code [6].

**3.1.4 Text Editor**

Sublime Text is a text editor that can be used for code, markup and prose. It is sophisticated and performing and it has a slick user interface that contains a range of different features [7].
### 3.2 Schedule

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Tasks to be performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 19th – January 23rd</td>
<td>- Project Selection</td>
</tr>
<tr>
<td>January 26th – January 30th</td>
<td>- Meeting with the Business Office</td>
</tr>
<tr>
<td></td>
<td>- Project Initial Specification</td>
</tr>
<tr>
<td>February 2nd – February 6th</td>
<td>- Feasibility Study and Analysis</td>
</tr>
<tr>
<td>February 9th – February 13th</td>
<td>- Capstone Diaries</td>
</tr>
<tr>
<td>February 16th – February 20th</td>
<td>- Project Design and Analysis</td>
</tr>
<tr>
<td>February 23rd – February 27th</td>
<td>- Capstone Diaries</td>
</tr>
<tr>
<td>March 2nd – March 6th</td>
<td>- Project Implementation</td>
</tr>
<tr>
<td>March 9th – March 13th</td>
<td>- Continuous Work on Interim Report</td>
</tr>
<tr>
<td>March 16th – March 20th</td>
<td></td>
</tr>
<tr>
<td>March 23rd – March 27th</td>
<td></td>
</tr>
<tr>
<td>March 30th – April 3rd</td>
<td></td>
</tr>
<tr>
<td>April 6th – April 10th</td>
<td></td>
</tr>
<tr>
<td>April 13th – April 17th</td>
<td>- Finalization of Final Report</td>
</tr>
<tr>
<td></td>
<td>- Testing and Debugging</td>
</tr>
<tr>
<td>April 20th – April 24th</td>
<td>- Final Report and Professional CV</td>
</tr>
<tr>
<td>April 27th – April 30th</td>
<td>- Project Defense</td>
</tr>
<tr>
<td>May 5th – May 6th</td>
<td>- Any corrections to report/project implementation finished</td>
</tr>
</tbody>
</table>

*Table 3.2: Project Schedule*
4 Design and Analysis

4.1 Entity Relationship Diagram

The entity relationship diagram of the database to be used by the system is depicted in the figure hereafter. It contains all the entities, attributes, primary and foreign keys and cardinalities.

Figure 4.1: ERD
4.2 Use Case Diagram

Figure 4.2: Use Case Diagram
4.3 Examples of Sequence Diagrams

4.3.1 Authentication Sequence Diagram

![Faculty Authentication Sequence Diagram](image1)

Figure 4.3.1: Faculty Authentication Sequence Diagram

4.3.2 Add Apartment Sequence Diagram

![Add Apartment Sequence Diagram](image2)

Figure 4.3.2: Add Apartment Sequence Diagram
4.3.3 Update Apartment Sequence Diagram

Figure 4.3.3: Update Apartment Sequence Diagram

4.3.4 Delete Apartment Sequence Diagram

Figure 4.3.4: Delete Apartment Sequence Diagram
Development and Implementation

5.1 Data Model Tier

Through this tier, the application communicates with the database. As mentioned in the feasibility study, MySQL is used as a database server for the project. Connecting to the database requires the creation of new DB connection using PHP. The following figure shows the piece of code used to establish the connection to the database.

![Figure 5.1: Connection to the Database](image)

5.2 Business Tier

The business logic constitute the tier of the logic behind the system. They contain the implementation of the methods that bridge the gap between the end user and the database. For example, the function in figure 5.2.1 retrieves the information of the user who is logged in the application to be displayed in his/her profile. Figure 5.2.2 shows the function that retrieves the information about the faculty member who is occupying the apartment. Figure 5.2.3 represents the update and delete apartment functions that are used by the Housing Department. These represent only some of the many functions that are used in the application.
Figure 5.2.1: User Profile Information
Figure 5.2.2: Faculty Occupying Apartment Information
Figure 5.2.3: Update and Delete Apartment Functions
5.3 Web Tier

Figure 5.3.1: Login Page

```html
<html>
<head>
    <title>Login Page</title>
</head>
<body>
    <div class="login-form">
        <form action="login_check.php" method="post">
            <input type="text" name="username" id="username" />
        </form>
    </div>
</body>
</html>
```

Figure 5.3.2: CSS Code

```css
html {
    margin: 0;
    padding: 0;
    background: url(pic/logo.png) no-repeat center fixed;
    -webkit-background-size: cover;
    background-size: cover;
}
header{
    background-color: #f4f4f4;
    border: 1px solid black;
    width: 100%;
    height: 100px;
    position: fixed;
    left: 0px;
    top: 0px;
}
header ul {
    list-style-type: none;
    width: 1000px;
    margin-top: 0px;
    margin-left: auto;
    margin-right: auto;
}
```
5.4 Screenshots of the Application

Figure 5.4.1: Login/First Page

```javascript
function slideshow(speed) {
  // append an 'li' item to the 'ul' list for displaying the caption
  $('ul.slideshow').append('<li id="slideshow-caption" class="caption"><div class="slideshow-caption-container"><p class="caption"></p><p class="caption"></p></div></li>');
  // set the opacity of all images to 0
  $('ul.slideshow li').css({opacity: 0.0});
  // get the first image and display it
  $('ul.slideshow li:.first').css({opacity: 1.0}).addClass('show');
  // get the caption of the first image from the 'rel' attribute and display it
  $('slideshow-caption p').html($('ul.slideshow li.show').find('img').attr('alt'));
  // display the caption
  $('slideshow-caption').css({opacity: 0.6, bottom:0});
  // call the gallery function to run the slideshow
  var timer = setInterval('gallery()',speed);
  // mouse the slideshow on mouse over
  $('ul.slideshow').hover(
    function () {
      clearInterval(timer);
    },
    function () {
      timer = setInterval('gallery()',speed);
    }
  );
}
```
Figure 5.4.2: Login Failure Page

Figure 5.4.3: Missing Username Page
Figure 5.4.4: Missing Password Page

Figure 5.4.5: Housing Home Page
Figure 5.4.6: Housing Apartments Page

Figure 5.4.7: Housing Chosen Free Apartment and Output of Free Function
Figure 5.4.8: Housing Occupied Apartment Chosen

Figure 5.4.9: Housing Chosen Apartment Update
Figure 5.4.10: Housing Chosen Apartment Delete

Figure 5.4.11: Housing Faculty Members
Figure 5.4.12: Housing Faculty Chosen

Figure 5.4.13 (There are two images because the page was too long to capture in one image): Housing Faculty Update
Figure 5.4.14: Housing New Home Page

Figure 5.4.15: Housing New Apartment Home Page
Figure 5.4.16 (There are two images because the page was too long to capture in one image): Housing New Faculty Home Page
Figure 5.4.17 (There are two images: Before and after): Profile Update
Figure 5.4.18: Password Change

Figure 5.4.19: Business Office Home Page
Figure 5.4.20: Business Office Apartments

Figure 5.4.21: Business Office Faculty
Figure 5.4.22: Business Office Charges

Figure 5.4.23: Business Office New Page
Figure 5.4.24: Business Office Add New Charge

Figure 5.4.25: Faculty Home Page
Figure 5.4.26: Faculty Water Bill

Figure 5.4.27: Faculty Water Bill Print
Figure 5.4.28: Faculty Electricity Bill

Figure 5.4.29: Faculty Electricity Bill Print
Figure 5.4.30: About Page 1 (First Image of SlideShow)

Figure 5.4.31: About Page 2 (Second Image of SlideShow)
6 Conclusion

Learning outcomes

This capstone project was of a great added value because first of all, it enabled me to work individually on a project, in parallel with taking four other classes, while we used to work on similar projects in team of three or four in other courses. Secondly, I learned something new, namely coding with PHP and JavaScript since they are not languages that we study at university, and I never worked with them before during any of my internships. That being said, I have reinforced many theoretical and technical skills that I have acquired throughout my undergraduate studies, especially on the programming and software engineering levels. In addition to this, this project went beyond the skills related to my major and allowed me to strengthen my management skills.

Difficulties

As every person living a new experience, I had one main difficulty related to this project. On the one hand, while I started coding using C# on the ASP.NET framework, which were my first chosen technology enablers, I encountered some difficulties in my local database established on Visual Studio on the lab computers. This latter wouldn’t download on my personal computer, so I had to look for another option in which IDEs and framework could be open source. I had two options: either work with JEE or learn PHP and use it. Since I already worked with JEE during my last internship, I decided to go for PHP and learn something new, especially that it is one of the most used languages in the professional work places. I encountered this problem a little bit late because it was during the last phase of my project, which is the implementation phase; therefore, I had to divide my time very tightly in order to learn PHP and make sure to finish the major functionalities of my application on time.

For the above reason, some special features like notifications could be left as a future work for this application, even if it actually fulfills all the requirements of my client. Despite all these difficulties, I do not regret any of the challenges I encountered because they enabled to learn many new things that did not belong to my plan.
References

http://formation.alterway.fr/


http://www.w3.org/community/webed/wiki/HTML

http://www.w3schools.com/css/css_intro.asp


http://www.sublimetext.com/
Appendix A

Example of Electricity Bill Provided by the Client:

Dear Client,

Please remit the amount of _______ - DH to the Office of Business Services no later than 15 days from the date of this invoice.

If you’d rather have this amount deducted from your salary, please let us know by sending an email to: m.aaquil@aui.ma, Cc e.rhannai@aui.ma.

<table>
<thead>
<tr>
<th>HISTORIC</th>
<th>January</th>
<th>February</th>
<th>March</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Due</td>
<td>Paid</td>
<td>Balance</td>
</tr>
<tr>
<td>277.13</td>
<td></td>
<td>621.20</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>April</th>
<th>May</th>
<th>June</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Due</td>
<td>Paid</td>
<td>Balance</td>
</tr>
<tr>
<td>192.56</td>
<td></td>
<td>192.56</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>July</th>
<th>August</th>
<th>September</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Due</td>
<td>Paid</td>
<td>Balance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>October</th>
<th>November</th>
<th>December</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Due</td>
<td>Paid</td>
<td>Balance</td>
</tr>
</tbody>
</table>

With Best Regards,